CS 2336 – Last Assignment

NOTES:

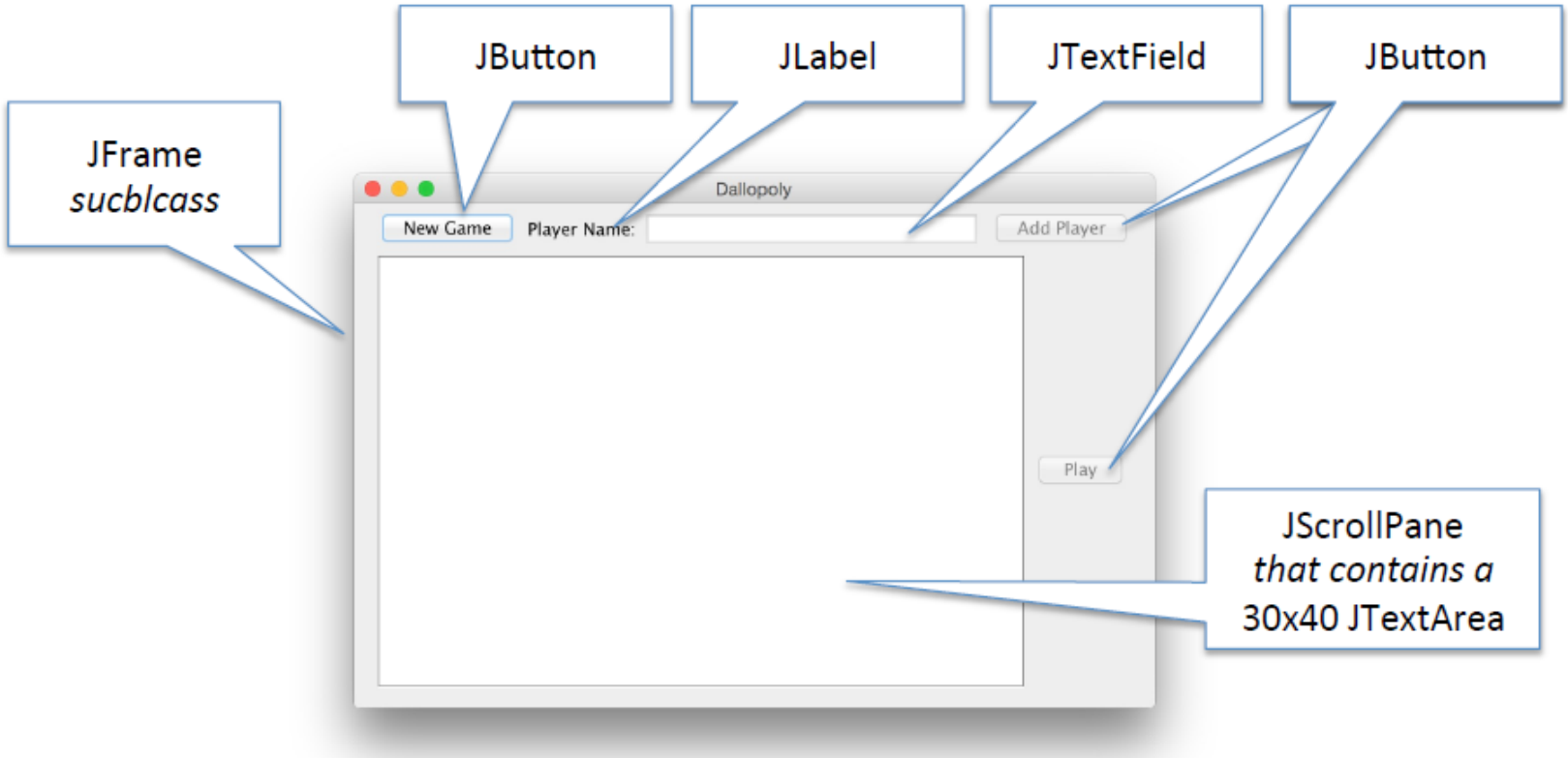
Each program should include comments that explain what each block of code is doing. Additionally, the programs should compile without errors, and run with the results described in the exercise. The following deductions will be made from each exercise if any of the following is incorrect or missing:

Proper formatting [5 points]
Proper names for classes and variables [5 points]
Comments [5 point]
Program doesn't compile [10 points]
Source code (java file) missing [10 points]
Executable (class file) missing [10 points]
Missing array where an array was required [5 points]
Missing loop where a loop was required [5 points]
Missing class from the design provided [10 points]
Missing method from the design provided [5 points]

Lab (100 Points)

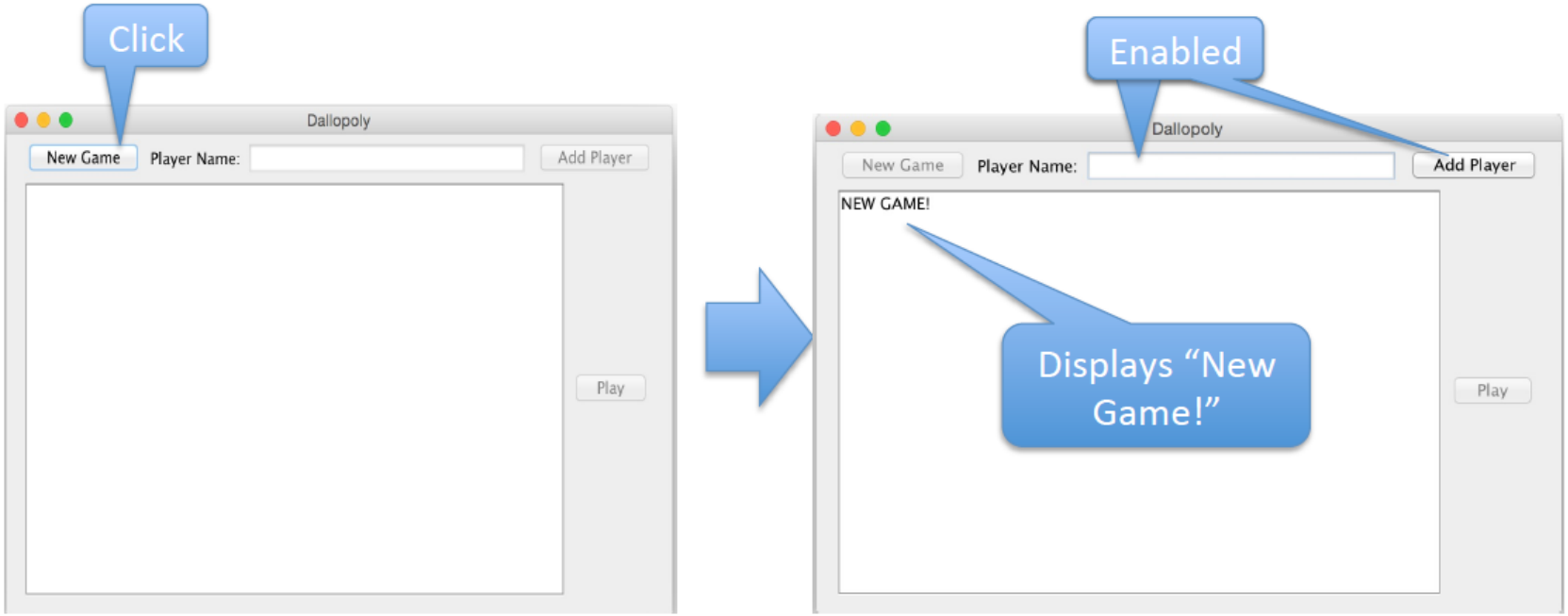
Add a user interface on top of the game you created in previous lab.

You must make some changes to your previous lab solution for this to work. Instead of printing game progress on the console, you will generate Strings that contain game information to display in the JTextArea of your user interface.



CREATING A NEW GAME

When you launch the Frame, all widgets are disabled except the “New Game” button. When “New Game” is clicked, an instance of Dallopoly is created, the “New Game” button is disabled, and the player name text field and “Add Player” buttons are enabled. Also, the message “New Game” is displayed in the JTextArea.

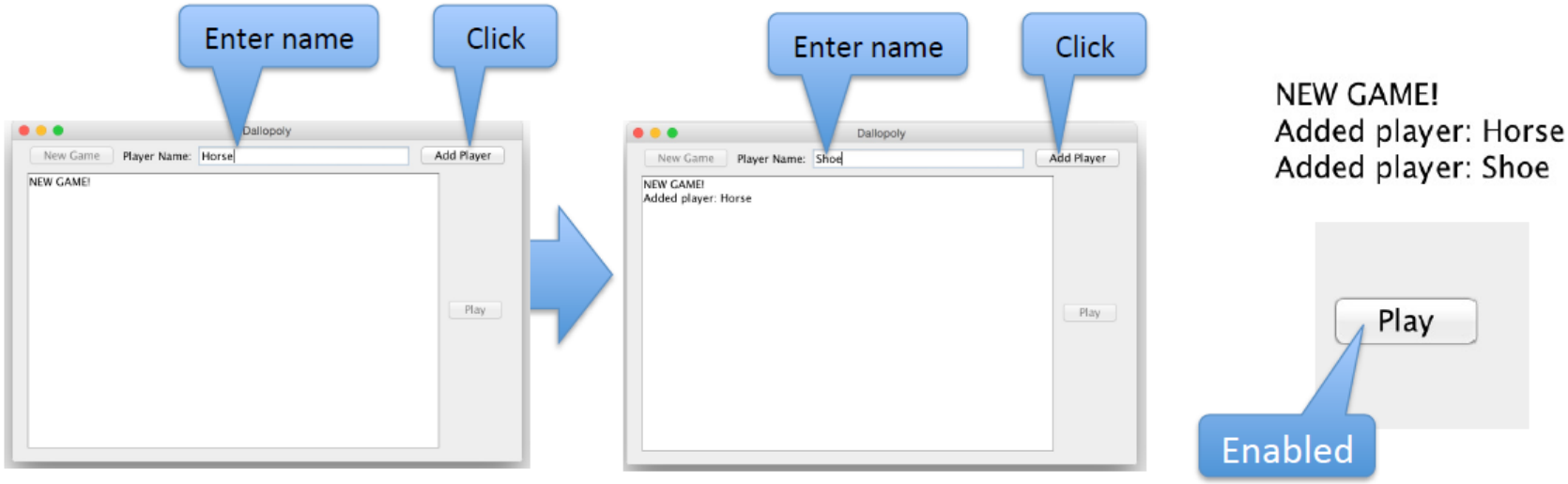


ADDING PLAYERS

Enter a name in the player name text field, and click the “Add Player” button.

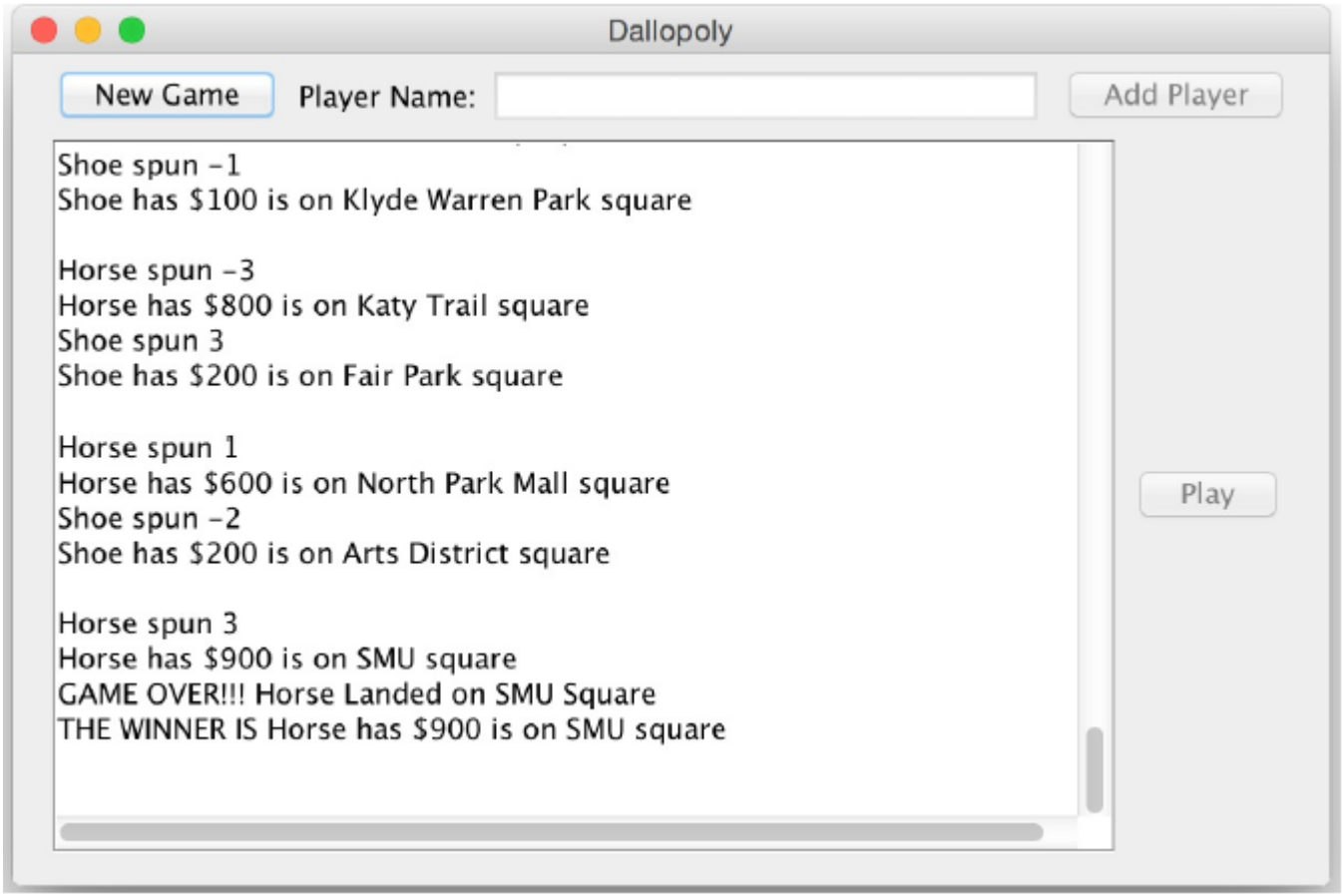
Send the addPlayer message to the game and print the new player in the JTextArea.

Clear the player name text field so another player can be added. Keep track of the number of players that are added. After a second player is added, enable the “Play” button.



PLAYING THE GAME

When the “Play” button is clicked, send the playGame message to the Dallopoly object. It should accumulate all the game progress in a large single String that is returned to the Frame. Using the JTextArea’s *append* method, put the game results content in the JTextArea, enable the “New Game” button and disable all the other widgets.



CHANGES TO YOUR PREVIOUS CODE:

Because the Frame requires Strings to populate the contents its JTextArea widget, some changes are required to the solution you built for previous lab. Instead of printing game results to the console, you will create a String in the *playGame* method which appends the ongoing progress of the game as a continuously concatenated String.

(1) Change the return type of Dallopoly's *playGame* from void to String
(2) Create a String variable at the beginning of *playGame.* Each time you print text to the console, replace this with code that appends the text to that String.
(3) When you send the *takeTurn* message to a player from within the *playGame* method, *takeTurn* should now return a String that you will also concatenate onto that String.
(4) Return the contents of the String at the end of the *playGame* method.
(5) Your JFrame will append the contents of the JTextArea with the contents of the String that is returned from the *playGame* method.
(6) To support the above steps, change the return type of Player's *takeTurn* method from *void* to String. Within *takeTurn*, create a String and concatenate all of the results of a player's turn onto the end of that String. Return the contents of the String to Dallopoly to add to its String.

continued…

CHANGES TO YOUR PREVIOUS LAB CODE (continued):

One additional change is required to your previous lab code solution for this version to work. You are no longer creating two players within the Dallopoly constructor. Remove this code from the Dallopoly constructor. Instead, add the following method to the DallopolyGame class:

```
public String addPlayer(String s){
    players.add(new Player(s, theBoard.getStartSquare()));
    return s;
}
```

In your JFrame, when the "Add Player" button is pressed, send the *addPlayer* message to the DallopolyGame object, passing the String contained in the player name JTextField to the *addPlayer* method shown above.

Summary of Contents

Class	Description
GameLauncher	Has *main* method that creates the JFrame named DallopolyWindow set to a size of 600x400
DallopolyWindow	JFrame subclass that includes all window widgets and behavior. In addition to variables for each of the window components, be sure to include variables to contain an instance of Dallopoly and an int to contain the total number of players that have been added. (Initialize this to zero.) It should have a DallopolyWindow constructor as well as an inner class to handle behavior.
Dallopoly	Contains the contents of Dallopoly you created in Lab 7 with the changes described in this document. Methods include: Dallopoly (constructor), addPlayer, and playGame
Player	Containts the contents of the Player you created in Lab 7, which changes described in this document. Methods include Player (constructor), takeTurn, chngeMoney, getters, setters, and toString.
Board Square LastSquare PenaltySquare PrizeSquare Spinner	All these classes are unchanged from Lab 7, but you will still need them as-is for Lab 8 to work.
squareList.txt	Unchanged from Lab 7, but you will still need it for Lab 8 to work.